

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently amended) A method for checkpointing an application,
2 comprising:
3 | ~~pre-linking~~dynamically linking an interceptor library into the application
4 | during a run-time invocation of the application, wherein the run-time invocation
5 | occurs after the application has been ~~compiled~~compiled and linked;
6 intercepting a function call produced by the application at the interceptor
7 library;
8 recording parameters of the function call to create a checkpoint that
9 includes information about the function call parameters;
10 making the function call;
11 receiving results of the function call; and
12 forwarding results of the function call back to the application.

- 1 2. (Original) The method of claim 1, further comprising creating a
2 checkpoint by:
3 stopping the application;
4 retrieving the recorded parameters;
5 saving the checkpoint data, including the recorded parameters, to
6 secondary storage; and
7 resuming the application.

1 3. (Original) The method of claim 2, further comprising using the
2 checkpoint to restore the application.

1 4. (Original) The method of claim 2, wherein saving the checkpoint data to
2 secondary storage involves saving the checkpoint data to a persistent storage.

1 5. (Original) The method of claim 2, wherein saving the checkpoint data to
2 secondary storage involves saving the checkpoint data in a file system, or a
3 database.

1 6. (Original) The method of claim 1, wherein making the function call
2 involves referencing the function through a function pointer.

1 7. (Original) The method of claim 1, further comprising recording the
2 results of the function call to facilitate creating a checkpoint that includes
3 information about the results of the function call.

1 8. (Original) The method of claim 1, wherein the function calls can include
2 system calls or lib calls.

1 9. (Original) The method of claim 1, wherein the parameters can include:
2 file paths;
3 thread flags; and
4 timer-thread relationships.

1 10. (Currently amended) A computer-readable storage medium storing
2 instructions that when executed by a computer cause the computer to perform a
3 method for checkpointing an application, the method comprising:

4 | ~~pre-linking~~dynamically linking an interceptor library into the application
5 | during a run-time invocation of the application, wherein the run-time invocation
6 | occurs after the application has been ~~compiled~~compiled and linked;
7 | intercepting a function call produced by the application at the interceptor
8 | library;
9 | recording parameters of the function call to create a checkpoint that
10 | includes information about the function call parameters;
11 | making the function call;
12 | receiving results of the function call; and
13 | forwarding results of the function call back to the application.

1 | 11. (Original) The computer-readable storage medium of claim 10, further
2 | comprising creating a checkpoint by:
3 | stopping the application;
4 | retrieving the recorded parameters;
5 | saving the checkpoint data, including the recorded parameters, to
6 | secondary storage; and
7 | resuming the application.

1 | 12. (Original) The computer-readable storage medium of claim 11, further
2 | comprising using the checkpoint to restore the application.

1 | 13. (Original) The computer-readable storage medium of claim 11,
2 | wherein saving the checkpoint data to secondary storage involves saving the
3 | checkpoint data to a persistent storage.

1 14. (Currently amended) The computer-readable storage medium of ~~claim~~
2 ~~12~~claim 11, wherein saving the checkpoint data to secondary storage involves
3 saving the checkpoint data in a file system, or a database.

1 15. (Original) The computer-readable storage medium of claim 10,
2 wherein making the function call involves referencing the function through a
3 function pointer.

1 16. (Original) The computer-readable storage medium of claim 10,
2 wherein the method further comprises recording the results of the function call to
3 facilitate creating a checkpoint that includes information about the results of the
4 function call.

1 17. (Original) The computer-readable storage medium of claim 10,
2 wherein the function calls can include system calls or lib calls.

1 18. (Original) The computer-readable storage medium of claim 10,
2 wherein the parameters can include:
3 file paths;
4 thread flags; and
5 timer-thread relationships.

1 19. (Currently amended) An apparatus that checkpoints an application,
2 comprising:
3 a ~~pre-linking~~dynamic linking mechanism that is configured to ~~pre-~~
4 ~~link~~linkdynamically link an interceptor library into the application during a run-time
5 invocation of the application, wherein the run-time invocation occurs after the
6 application has been ~~compiled~~compiled and linked;

7 an intercepting mechanism within the interceptor library that is configured
8 to intercept a function call produced by the application;
9 a recording mechanism that is configured to record parameters of the
10 function call to facilitate creating a checkpoint that includes information about the
11 function call parameters;
12 a calling mechanism that is configured to make the function call;
13 a receiving mechanism that is configured to receive results of the function
14 call; and
15 a forwarding mechanism that is configured to forward results of the
16 function call back to the application.

1 20. (Original) The apparatus of claim 19, further comprising a checkpoint
2 creation mechanism that is configured to:
3 stop the application;
4 retrieve the recorded parameters;
5 save the checkpoint data, including the recorded parameters, to secondary
6 storage; and to
7 resume the application.

1 21. (Original) The apparatus of claim 20, further comprising a restoration
2 mechanism that is configured to use the checkpoint data to restore the application
3 to the checkpointed state.

1 22. (Original) The apparatus of claim 20, wherein the checkpoint creation
2 mechanism is configured to save checkpoint data to a persistent storage.

1 23. (Original) The apparatus of claim 20, wherein the checkpoint creation
2 mechanism is configured to save the checkpoint data in a file system, or a
3 database.

1 24. (Original) The apparatus of claim 19, wherein the calling mechanism
2 is configured to make the function call by referencing the function through a
3 function pointer.

1 25. (Original) The apparatus of claim 19, further comprising a recording
2 mechanism that is configured to record the results of the function call to facilitate
3 creating a checkpoint that includes information about the results of the function
4 call.

1 26. (Original) The apparatus of claim 19, wherein the function calls can
2 include system calls or lib calls.

1 27. (Original) The apparatus of claim 19, wherein the parameters can
2 include:
3 file paths;
4 thread flags; and
5 timer-thread relationships.